On The Successive Projections Approach
To
Least-Squares Problems

by

J.E. Dennis, Jr.[1]

and

Trond Steihaug[2]

Technical Report 83-18, August 1983
(Revised February 1985)

Report Documentation Page

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **FEB 1985**	2. REPORT TYPE	3. DATES COVERED **00-00-1985 to 00-00-1985**

4. TITLE AND SUBTITLE **On the Successive Projections Approach to Least-Squares Problems**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Computational and Applied Mathematics Department ,Rice University,6100 Main Street MS 134,Houston,TX,77005-1892**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **29**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Abstract

In this paper, we suggest a generalized Gauss-Seidel approach to sparse linear and nonlinear least-squares problems. The algorithm, closely related to one given by Elfving (1980), uses the work of Curtis, Powell, and Reid (1974) as extended by Coleman and Moré (1983) to divide the variables into nondisjoint groups of structurally orthogonal columns and then projects the updated residual into each column subspace of the Jacobian in turn. In the linear case, this·procedure can be viewed as an alternate ordering of the variables in the Gauss-Seidel method. Preliminary tests indicate that this leads quickly to cheap solutions of limited accuracy for linear problems, and that this approach is promising for an inexact Gauss-Newton analog of the inexact Newton approach of Dembo, Eisenstat, and Steihaug (1981).

1. Introduction

The purpose of this paper is to suggest a generalized group Gauss-Seidel approach to sparse least-squares problems that appears in preliminary tests to be useful in obtaining cheap solutions of limited accuracy. For nonlinear problems, the approach combines the work on linearization of Coleman and Moré (1982), (1983) developing ideas of Curtis, Powell, and Reid (1974) with the work on inexact Newton methods of Dembo, Eisenstat, and Steihaug (1982) and Steihaug (1980). This amalgam leads here to an especially convenient implementation of the inexact Gauss-Newton method suggested in Section 3.

The basic idea is simple. If we apply the methods of estimating sparse Jacobian matrices to group the columns of a coefficient matrix, then each group of columns is mutually orthogonal in their zero/nonzero structure regardless of the particular values of the nonzero entries. This has the advantage of leading for each group of columns to smaller cheaper least-squares problems unaffected by conditioning. The idea is to cycle through these groups solving each time with the right-hand side updated as in iterative refinement.

These graph-theoretic algorithms actually partition the columns of the coefficient matrix A into disjoint groups. Notice that there is no reason to want nonintersecting groups of columns, and it will be interesting in our context to consider overlap between groups. We report some experiments in Section 4 with a simple heuristic scheme for overlapping orthogonal sets of columns and for groups whose normal matrices are tridiagonal.

In the nonlinear case, there seem to be diagnostic advantages to maximal overlap in the computation of finite-difference Jacobians for inaccurate residuals. This is a potentially useful point that has not to our knowledge been pointed out and applies equally in the solution of square systems of nonlinear equations. It would require no additional vector function evaluations to use a different perturbation for a given variable or column for each group to which it belongs. These values could then be used heuristically to refine the partial derivative estimate and estimate its accuracy. In Section 4, we give some evidence to support the obvious conjecture that overlapping orthogonal sets of columns is helpful in

the nonlinear case; although it does not seem worth the extra arithmetic in the linear case.

It is in order to disclaim that we bypass the effects of ill-conditioning by using any of the grouping strategies suggested here; although, we may make less use than factorization techniques of extended precision in obtaining a given accuracy. It is true that some of the subproblems we solve are unaffected by conditioning because they are diagonal, but a problem with two nearly-dependent columns shows what happens in the ill-conditioned case and illustrates the algorithm in residual space. We first project the right-hand side, say r^0, onto the subspace spanned by the first column a_1. Now we subtract from r^0 the part of the residual accounted for by this projection, and then project the new residual r^1 onto a_2. The algorithm repeats the process and the reader will see that in order to reduce to an acceptable level the part of the residual that can be accounted for by a linear combination of a_1 and a_2, we will need more iterations for a small angle between a_1 and a_2. If a_1 were orthogonal to a_2, then one projection onto each would suffice to reduce the residual as much as possible. This is exactly the feature that we are exploiting within each column group.

Since our algorithm is based on convenient column groupings, the reader might be tempted to think of this as a disadvantage over row-oriented schemes for large least-squares problems. This is not the case either here or in the nonlinear equations problem for that matter. See Coleman and Moré (1983), p.208, for a discussion that just the opposite is likely to be the case. There are a host of schemes that can be used to adapt the ideas here to any segmentation of the problem into groups of rows with the elements in each segment stored columnwise for which the corresponding rows of the residuals can be calculated separately.

In Section 2, we will present the algorithm for the linear problem and show that it generalizes the group SOR method in the sense that it is that method for the normal equations of an extended linear least-squares problem. This observation will allow an easy convergence proof, even in the singular case, using results from Keller (1965). Section 3 is a discussion of the nonlinear least-squares problem that combines the algorithm of Section 2

applied to the linearized problem with the inexact Gauss-Newton approach as a guide to the accuracy required in solving the linearized problems. Section 4 presents some numerical results for several heuristic column grouping schemes as well as for the associated orderings of the columns for point SOR.

2. The Algorithm for the Linear Least Squares Problem

Let A be an m by n real matrix, $m \geq n$, $b \in R^m$, and consider the least-squares problem

$$\min_{x \in R^n} \| Ax - b \|_2 . \tag{2.1}$$

In order to illustrate our algorithm, assume that the columns of A are divided into g groups $A_1, A_2, \ldots, A_g$, where A_i is a m by n_i submatrix and A_i may share columns with A_j. Let $x_1 \in R^{n_1}$, $x_2 \in R^{n_2}$, $\ldots$, $x_g \in R^{n_g}$. The least-squares problem (2.1) can now be written as:

$$\min \{ \|A_1 x_1 + A_2 x_2 + \ldots + A_g x_g - b\|_2 : x_i \in R^{n_i}, i=1,2,\ldots,g \}. \tag{2.2a}$$

Note that (2.2a) is really an m by $\mathbf{n} = n_1 + n_2 + \ldots + n_g$ least-squares problem which we will denote using boldface type as:

$$\min_{x \in R^n} \| \mathbf{A}x - b \|_2 , \tag{2.2b}$$

where $\mathbf{A} = (A_1 | A_2 | \ldots | A_g)$ is an $m \times \mathbf{n}$ matrix and $\mathbf{x} = (x_1, x_2, \ldots, x_g)^T$ is a $\mathbf{n}$ vector. Clearly, $\mathbf{A}$ has exactly the same set of distinct columns as A divided into the same groups, but it will be useful that we can ignore overlaps if we view the A_i as column groups of $\mathbf{A}$. It will also be convenient to have the notation $\overline{x}_i \in R^n$ and $\overline{\mathbf{x}}_i \in R^n$ to denote respectively the vectors obtained by starting with the n or $\mathbf{n}$ zero vectors and placing the nonzero entries of x_i in the positions corresponding to the column indices in A or $\mathbf{A}$ of A_i. Thus, given either $\mathbf{x} \in R^n$, or $x_1, x_2, \ldots, x_g$ with each $x_i \in R^{n_i}$, we can define $x \in R^n$, $x = \overline{x}_1 + \overline{x}_2 + \cdots + \overline{x}_g$ and write

$$Ax = A\overline{x}_1 + A\overline{x}_2 + \cdots + A\overline{x}_g = A_1 x_1 + A_2 x_2 + \cdots + A_g x_g$$

or

$$x = \bar{x}_1 + \bar{x}_2 + \cdots + \bar{x}_g, \quad Ax = A\bar{x}_1 + A\bar{x}_2 + \cdots + A\bar{x}_g = A_1 x_1 + A_2 x_2 + \cdots + A_g x_g.$$

Suppose we have an approximation x^k to a solution x^* to (2.1), and we divide x^k into $x_1^k, x_2^k, \ldots, x_g^k$ as above. (This division will not be unique for components corresponding to column overlaps.) Then (2.2) suggests the following successive replacements iteration.

FOR $i=1,2,\ldots,g$ DO

Solve for x_i^{k+1} : $\min\{ \|\sum_{j=1}^{i} A_j x_j^{k+1} + \sum_{j=i+1}^{g} A_j x_j^k - b\|_2 : x_i^{k+1} \in R^{n_i}\}.$

This is a method of projections [Householder (1964)], and for the special case of non-overlapping column groups, i.e., for formulation (2.2b), this particular iteration was suggested by Elfving (1980). We will see easily below that this is group Gauss-Seidel [Young (1971),p.438] on the normal equations for (2.2) and so it is a generalized group Gauss-Seidel for (2.1). Björk and Elfving (1979) and Elfving (1980) have pointed out that in this form of the Gauss-Seidel iteration, we do not need to explicitly form the normal equations.

Let s^k be the step or correction, let $r^k = Ax^k - b$ be the residual, and notice that

$$A_1 x_1^{k+1} + \sum_{j=2}^{g} A_j x_j - b = A_1 s_1^k + r^k.$$

So we rewrite the iteration:

FOR $i=1,2,\ldots,g$ DO

Solve for s_i^k : $\min \{ \|A_i s_i^k + r^{k+(i-1)/g}\| : s_i^k \in R^{n_i} \}$;

Update the residual: $r^{k+i/g} = r^{k+(i-1)/g} + A_i s_i^k.$

The new approximate solution is now

$$x_i^{k+1} = x_i^k + s_i^k, \, i=1,\ldots,g; \quad x^{k+1} = \sum_{i=1}^{g} \bar{x}_i^{k+1}.$$

We complete this section by stating the general algorithm with relaxation factors and proving that it converges. In Section 4, we will discuss termination criteria and storage requirements. We will want to assume that each A_i has full rank. This can be done without any loss of generality, since any linearly dependent group of columns can be split into smaller groups of linearly independent columns by making each column a group by itself, if

necessary. Clearly, we can assume that there is no zero column, since such a column can be dropped from A without changing the least-squares problem.

Subdivide A into g groups. (Each A_i has full rank, $i=1,2,\cdots,g$.)

Choose x_i^0, $i=1,\cdots,g$.

Compute $r^0=Ax^0-b$. (Choose $0<\omega_i<2$, $i=1,2,\cdots,g$.)

FOR $k=0$ STEP 1 UNTIL Convergence DO

    FOR $i=1$ STEP 1 UNTIL g DO

      $s_i^k = -(A_i^T A_i)^{-1} A_i^T \, r^{k+(i-1)/g};$

      $r^{k+i/g} = r^{k+(i-1)/g} + \omega_i A_i s_i^k;$

      $x^{k+i/g} = x^{k+(i-1)/g} + \omega_i \bar{s}_i^k;$

    Check for Convergence.

Theorem 2.1: Let the columns of A be divided into g groups $A_1,\cdots,A_g$ and let each A_i have full rank. Let $\{x^k\}$ be generated by the above algorithm with any choices $0<\omega_i<2$, and any $x_i^0 \in \mathbf{R}^{n_i}$, $i=1,2,\cdots,g$. Then $\{x^k\}$ converges to x^*, a solution to the least-squares problem (2.1).

Proof: We will show that the algorithm is the group SOR iteration on the normal equations for (2.2). We will then apply a result of Keller (1965) which proves convergence for the group SOR method applied to positive semi-definite systems. This will give convergence of $\{\mathbf{x}^k\}$ and hence of $\{x_i^k\}$ and of $\{\bar{x}_i^k\}$ for every i. But then $\{x^k\}$ converges since it is just $\{\sum_{i=1}^{g} \bar{x}_i^k\}$.

We now state the result of Keller (1965) for completeness. Let G be a real symmetric matrix of order n of the form

$$G = D + E + E^T \qquad (2.3)$$

and let W be any real nonsingular matrix such that

$$N = W^{-1}D + E \qquad (2.4)$$

is nonsingular. Let f be any vector for which the system

$$Gx = f \qquad (2.5)$$

has a solution. Consider the iterative method

$$Nx^{k+1} = (N-G)x^k = f \qquad (2.6)$$

where x^0 is an initial guess of a solution of (2.5). The following lemma is a part of Corollary 2.1 of Keller (1965).

Lemma 2.2. Let G be a symmetric positive semidefinite matrix of the form (2.3) and let W be nonsingular such that the matrix N in (2.4) is nonsingular. Let (2.5) have a solution and let

$$P = W^{-1}D + (W^{-1}D)^T - D$$

be positive definite. Then for every x^0 the sequence $\{x^k\}$ of (2.6) converges to a solution of (2.5).

It will be useful to let $G_{ij}=A_i^T A_j$ be the (i,j) block of the n by n Gram matrix, $G=\mathbf{A}^T\mathbf{A}$. Define

$$x_j^k = x_j^0 + \sum_{p=0}^{k-1} \omega_j s_j^p \quad j=1,2,\cdots,g, \quad \text{and} \quad x^{k+i/g} = \sum_{j=1}^{g}\bar{x}_j^k + \sum_{j=1}^{i-1}\omega_j \bar{s}_j^k .$$

First we need that

$$r^{k+i/g} = Ax^{k+i/g}-b.$$

We will prove this by induction on $k+i/g$ in steps of $1/g$. By definition, the statement is true for $k+i/g=0$. Now assume that the statement is true for some $k+(i-1)/g \geqq 0$. Then

$$r^{k+i/g} = r^{k+(i-1)/g} + \omega_i A_i s_i^k = Ax^{k+(i-1)/g} - b + \omega_i A \bar{s}_i^k$$

$$= A[x^{k+(i-1)/g} + \omega_i \bar{s}_i^k] - b = Ax^{k+i/g}-b.$$

Now,

$$G_{ii}[x_i^{k+1} - x_i^k] = \omega_i G_{ii} s_i^k = -\omega_i A_i^T[Ax^{k+(i-1)/g} - b]$$

$$= - \omega_i [\sum_{j=1}^{i-1}G_{ij}x_j^{k+1} + \sum_{j=i}^{g}G_{ij}x_j^k - A_i^T b],$$

which becomes:

$$G_{ii}x_i^{k+1} + \omega_i[\sum_{j=1}^{i-1}G_{ij}x_j^{k+1} + \sum_{j=i+1}^{q}G_{ij}x_j^k] + (\omega_i-1)G_{ii}x_i^k = \omega_i A_i^T b.$$

When $\omega_i=\omega, i=1,2,...,g$ this is the standard form given on p. 438 of Young (1971) of the group SOR method applied to $Gx=A^T b$. To apply Keller's result, we write $G=D+E+E^T$ where $D=(G_{ii})$ is the n×n block diagonal of G, and $E=(G_{ij})$ is the n×n block strict lower triangle of G. Let W be the n×n block diagonal matrix whose ith block is ω_i times the $n_i \times n_i$ identity matrix. Now we rewrite the iteration as

$$D\mathbf{x}^{k+1} + W[E\mathbf{x}^{k+1} + E^T\mathbf{x}^k] + (W-I)D\mathbf{x}^k = WA^T b$$

$$(D+WE)\mathbf{x}^{k+1} + [W(D+E^T)-D]\mathbf{x}^k = WA^T b.$$

Since no $\omega_i=0$, we can multiply through by W^{-1}, set $N=W^{-1}D+E$, and obtain

$$N\mathbf{x}^{k+1} + (G-N)\mathbf{x}^k = A^T b.$$

In order to complete the proof, we only need that N is nonsingular and that $(2W^{-1}-I)D$ is positive definite. Observe that, since each A_i has full rank, D is positive definite. It follows that $W^{-1}D$ is nonsingular and so N is also. In fact, W^{-1} is blocked into constant diagonal matrices so that $0<\omega_i<2$ ensures that $(2W^{-1}-I)D$ is symmetric and positive definite.

3. The Inexact Gauss-Newton Approach

We now consider the algorithm for the nonlinear least squares problem. Let

$$F: \mathbf{R}^n \to \mathbf{R}^m, m \geq n, \text{ and } F = (F_1, \ldots, F_m)^T,$$

and define

$$\varphi(x) = \frac{1}{2} F(x)^T F(x).$$

Then the nonlinear least squares problem is to find x^* so that for some $\varepsilon>0$,

$$\varphi(x^*) \leq \varphi(x) \text{ for all } ||x-x^*|| < \varepsilon. \tag{3.1}$$

The simultaneous solution of n nonlinear equations in n unknowns may be viewed as solving (3.1) where $m=n$. For small-residual sparse problems, the Gauss-Newton method is very attractive. It starts with x^0 and generates a sequence of iterates $\{x^k\}$ as follows:

FOR $k=0$ STEP 1 UNTIL Convergence DO

$\qquad$ Solve for s^k: min$\{ \| F'(x^k)s + F(x^k)\|_2 : s \in \mathbb{R}^n \}$; $\qquad$ (3.2)

$\qquad$ Set $x^{k+1} = x^k + s^k$.

If $m \times n$ is large, solving the linearized problem (3.2) may require the techniques mentioned above for (2.1).

If we use an iterative method to solve the linearized problem, then it is important to know how accurately it must be solved in order to not impede convergence. We define the *Inexact Gauss-Newton* algorithm for a given real non-negative sequence $\{\theta_k\}$ and starting point x^0 as follows:

FOR $k=0$ STEP 1 UNTIL Convergence DO

$\qquad$ Find *some* approximate minimizer s^k of (3.2) so that

$$\frac{\|F'(x^k)^T r^k\|}{\|F'(x^k)^T F(x^k)\|} \leq \eta_k, \text{ where } r^k = F(x^k) + F'(x^k)s^k; \qquad (3.3)$$

$\qquad$ Set $x^{k+1} = x^k + s^k$.

where $\|\cdot\|$ is a vector norm or a consistent matrix norm. Let x^* be a solution of (3.1) and define the norm $\|y\|_* = \|F'(x^*)^T F'(x^*)y\|$. We will assume in this section that:

A.1. F is continuously differentiable in an open neighborhood Ω of x^*;

A.2. $F'(x^*)^T F(x^*) = 0$;

A.3. $F'(x^*)$ has full rank;

A.4. There exists $\gamma \geq 0$ so that for $x \in \Omega$,

$$\|(F'(x) - F'(x^*))^T F(x^*)\| \leq \gamma \|x - x^*\|_* . \qquad (3.4)$$

Notice that A.2 is somewhat redundant since we are assuming that x^* solves (3.1). Also, if the Jacobian matrix is Lipschitz continuous at x^*, i.e., there exists $L \geq 0$ so that for $x \in \Omega$,

$$\|F'(x^*) - F'(x)\|_2 \leq L\|x - x^*\|_2;$$

then there exists $\gamma \geq 0$ so that assumption A.4 holds in the l_2 norm. Notice that $\gamma = 0$ for zero-residual problems. The following theorem relates $\{\eta_k\}$ to the speed of convergence of

$\{x^k\}$.

Theorem 3.1: Assume that

$$\gamma + \eta(1+\gamma) < r < 1 \tag{3.5}$$

for γ from (3.4) and $0 \leq \theta_k \leq \eta$ from (3.3). Then there exists some $\varepsilon > 0$ for which

$\|x^0 - x^*\|_* \leq \varepsilon$ implies that the sequence of inexact Gauss-Newton iterates $\{ x^k \}$ is defined

and converges at least linearly to x^* in the sense that

$$\|x^{k+1} - x^*\|_* \leq r\|x^k - x^*\|_*.$$

Proof: Let

$$\mu = 2 \max \{ \|F'(x^*)^T\|, \|(F'(x^*)^T F'(x^*))^{-1}\| \} \tag{3.6}$$

and let $\delta > 0$ be so that

$$(1+\mu\delta)\Big[\eta[1+\gamma+(\mu+1)\delta]+\gamma+\mu\delta\Big] \leq r.$$

This can be done in view of (3.5). Define

$$G(x) \equiv F'(x)^T F'(x), \quad G^* \equiv F'(x^*)^T F'(x^*).$$

Choose $\varepsilon > 0$ so that if $\|x - x^*\|_* \leq \varepsilon$, then $\|G^{-1}(x)\| \leq \mu$, $\|G^* - G(x)\| \leq \delta$, $\|F'(x)^T\| \leq \mu$,

and $\|F(x^*) - F(x) - F'(x)(x^* - x)\| \leq \delta\|x - x^*\|_*$. This can be done in view of (3.6) and the

assumptions A.1 and A.3. Let x^+ be the new inexact Gauss-Newton iterate, i.e. x^+ satisfies

$$\frac{\|F'(x)^T r\|}{\|F'(x)^T F(x)\|} \leq \eta, \text{ where } r = F(x) + F'(x)s, \quad x^+ = x + s.$$

Consider

$$G^*(x^+ - x^*) = [(G^* - G)G^{-1} + I] [F'(x)^T r - (F'(x) - F'(x^*))^T F(x^*)$$
$$+ F'(x)^T(F(x^*) - F(x) - F'(x)(x^* - x))].$$

Taking norms yields

$$\|x^+ - x^*\|_* \leq \Big[1 + \|G(x)^{-1}\| \|G^* - G(x)\|\Big]$$
$$\times \Big[\|F'(x)^T r\| + \|(F'(x) - F'(x^*))^T F(x^*)\| + \|F'(x)^T\| \|F(x^*) - F(x) - F'(x)(x^* - x)\|\Big]$$
$$\leq (1+\mu\delta)\Big[\eta\|F'(x)^T F(x)\| + \gamma\|x - x^*\|_* + \mu\delta\|x - x^*\|_*\Big].$$

Consider

$$F'(x)^T F(x) = (F'(x) - F'(x^*))^T F(x^*) - F'(x)^T (F'(x^*) - F(x) - F'(x)(x^* - x))$$
$$- (G(x) - G^*)(x^* - x) - G^*(x^* - x).$$

Taking norms,

$$\|F'(x)^T F(x)\| \leq \gamma \|x^* - x\|_* + \mu\delta \|x^* - x\|_* + \delta \|x^* - x\|_* + \|x^* - x\|_*.$$

So

$$\|x^+ - x^*\|_* \leq (1 + \mu\delta) \Big[\eta[1 + \gamma + (\mu + 1)\delta] + \gamma + \mu\delta \Big] \|x - x^*\|_* \leq \tau \|x - x^*\|_*.$$

If $\gamma = 0$, then the inexact Gauss-Newton method is closely related to the inexact Newton method of Dembo, Eisenstat, and Steihaug (1982) and the inexact quasi-Newton method of Steihaug (1980). In the case when $\theta_k \equiv 0$, this theorem relaxes the Dennis (1977) conditions for convergence of the Gauss-Newton method. Furthermore, if F is twice continuously differentiable, then we can apply the standard Ostrowski Theorem to the Gauss-Newton method as in Ortega (1972). Now we will show that if the assumptions in the standard Ostrowski Theorem are satisfied, then (3.4) holds.

In the following discussion, let F be twice continuously differentiable in an open neighborhood Ω containing x^*. For x sufficiently close to x^*, A.3 lets us define

$$N(x) = x - \Big[F'(x)^T F'(x) \Big]^{-1} F'(x)^T F(x).$$

Then the derivative of N exists, is continuous in a neighborhood of x^* and

$$N'(x^*) = - \Big[F'(x^*)^T F'(x^*) \Big]^{-1} S$$

where

$$S = \sum_{i=1}^{m} F_i(x^*) \nabla^2 F_i(x^*)$$

(See 8.1.8 in Ortega, 1972, and Dennis, 1977.) Recall that the Gauss-Newton method is $x^{k+1} = N(x^k)$ and x^* is a point of attraction of the the Gauss-Newton iteration if $\rho(N'(x^*)) < 1$ where $\rho(\cdot)$ is the spectral radius of the matrix (see 8.1.7 in Ortega, 1972). Define the function $h:\Omega \rightarrow \mathbf{R}^n$ by

$$h(x) = (F'(x) - F'(x^*))^T F(x^*).$$

The assumption that F is twice continuously differentiable in an open neighborhood of x^*

and the assumption A.2 give $h'(x^*) = S$. Assume for some positive δ we have

$\rho(N'(x^*))+\delta \leq \gamma < 1$. Choose a norm $\|\cdot\|$ so that $\|N'(x^*)^T\| \leq \rho(N'(x^*))+\delta/2$. We can find

a vector norm that is consistent with the chosen matrix norm, and choose a neighborhood

of radius ε so that for all $\|x-x^*\| \leq \varepsilon$ we have

$$\|h(x)-h(x^*) - h'(x^*)(x-x^*)\| \leq \frac{\delta}{2} \|x-x^*\|.$$

This can be done since h is continuously differentiable. Consider

$$h(x) = h'(x^*)(x-x^*) + h(x)-h(x^*) - h'(x^*)(x-x^*)$$

and note that $h'(x^*)(x-x^*) = -N'(x^*)^T[F'(x^*)^TF'(x^*)(x-x^*)]$. Taking norms yields

$$\|(F'(x)-F'(x^*))^TF(x^*)\| = \|h(x)\| \leq \|h'(x^*)(x-x^*)\| + \|h(x)-h(x^*)-h'(x^*)(x-x^*)\|$$

$$\leq \left[\|N'(x^*)^T\| + \frac{\delta}{2} \right] \|x-x^*\|.$$

$$\leq (\rho(N'(x^*))+\delta) \|x-x^*\| \leq \gamma \|x-x^*\|.$$

which shows (3.4).

In the inexact Gauss-Newton approach, we ignore the specific method we are using to

find an approximate minimizer s^k of (3.2). If F' is sparse, then as in Curtis, Powell, and

Reid (1974), and Coleman and Moré (1982), (1983), we may group the columns of F' so

that the columns in each group are mutually orthogonal vectors. We note that a column

can be in several groups. The columns $F'(x^k)_i$ in group i may be approximated by finite

differences $\Delta F(x^k)_i$ with only one extra value $F(x^k+v_i)$, where v_i is an appropriate linear

combination of the corresponding standard unit vectors. For $s_i \in R^{n_i}$, let $\bar{s}_i \in R^n$ be con-

structed as in Section 2. This suggests the following cycle in the inner loop:

For given x^k, let $r^k = F(x^k)$, $y^k = x^k$;

Inner loop cycle:

FOR $i=1$ STEP 1 UNTIL g DO

      Compute $A_i^k = F'(x^k)_i$ or $\Delta F(x^k)_i$;

      Solve for s_i^k: $\min\{\|A_i^k s_i + r^{k+(i-1)/g}\|_2 : s_i \in \mathbf{R}^{n_i}\}$;

      Set $r^{k+i/g} = r^{k+(i-1)/g} + \omega_i A_i^k s_i^k$;

      Set $y^{k+i/g} = y^{k+(i-1)/g} + \omega_i \bar{s}_i^k$;

Check termination (3.3).

The next iterate is now $x^{k+1} = y^{k+c} \in \mathbf{R}^n$ where c is the number of cycles, which corresponds to terminating the inner iteration after c sweeps through each of the column groups. The least-squares problem (3.6) is trivial to solve when the columns in this group are mutually orthogonal. This especially convenient way to group the columns has been discovered independently by Coleman (1984).

If $c>1$, then the above inner loop cycle requires either recomputing $F'(x^k)_i$ or $\Delta F(x^k)_i$ when needed, or storing $F'(x^k)$ or $\Delta F(x^k)$. An alternative approach recomputes the Jacobian matrix of one particular group at a time and updates the nonlinear residual. This would suggest the following nonlinear substitution method:

Given x^0, compute $F(x^0)$

FOR $k=0$ STEP 1 UNTIL Convergence DO

FOR $i=1$ STEP 1 UNTIL g DO

      Compute $A_i^{k+(i-1)/g} = F'(x^{k+(i-1)/g})_i$ or $\Delta F(x^{k+(i-1)/g})_i$;

      Solve for s_i^k : $\min\{\|A_i^{k+(i-1)/g} s_i + F(x^{k+(i-1)/g})\|_2 : s_i \in \mathbf{R}^{n_i}\}$;

      Set $x^{k+i/g} = x^{k+(i-1)/g} + \omega_i \bar{s}_i^k$;

Check convergence.

4. Numerical Results

In this section, we describe two column grouping strategies to be used with the algorithms given in Sections 2 and 3, and we present some numerical results for the Duff and

Reid (1979) sparse least-squares test problems. We begin with a discussion of the problems.

4.1. The Test Problems

These problems are specified only in their sparsity structures which come from adjustment of survey data (Matrix 28 to 32 in the test bed).

Problem 1: A is 219 by 85 and the survey pattern is from Holland.

Problem 2: A is 958 by 292 and the survey pattern is from United Kingdom.

Problem 3: A is 331 by 104 and the survey pattern is from Scotland.

Problem 4: A is 608 by 188 and the survey pattern is from England.

Problem 5: A is 313 by 176 and the survey pattern is from Sudan.

The specific problems used here were found by generating the nonzero matrix elements randomly in the interval $(-1,1)$ and the components of a solution vector x randomly in the interval $(0,1)$. The righthand side b was found by computing $b=Ax$. The nonlinear problems were found by replacing x_j by x_j^3, i.e., component i in F is

$$F_i(x) = \sum_{j=1}^{n} A_{ij} x_j^3 - b_i.$$

Thus, our problems have zero residuals at the solution. We approximate all derivatives by finite differences in these tests.

4.2. The Column Grouping Schemes

We have already mentioned that one grouping scheme is based primarily on the ideas of Curtis, Powell, and Reid (1974) as expanded and improved by Coleman and Moré (1983). A FORTRAN code found in Coleman and Moré (1982) furnished our first pass partitioning the columns of A into disjoint groups. We will refer to this work as 'CM'. In Section 1, we argued that there could be some advantages to allowing the groups to overlap in some columns. In our tests, we used the following heuristic to expand each group in turn. To expand a given group, we first mark all columns that have a nonzero element in the

same row position as some column in the group. This identifies the columns that can not be added to the group. We then add one unmarked column to the group and add to the set of marked columns all columns that have a nonzero row element in the same position as the column that was added to the group. This process is then repeated until no columns are left unmarked.

Finally, let A_i denote a resulting submatrix of columns a_j of A, $j \in I_i$, then $A_i^T A_i$ is a diagonal matrix where the diagonal elements are the squares of the l_2-norms of the columns, so A_i has full rank, as we required in Theorem 2.1. As an example, we present in Table 1 the results of this scheme applied to Problem 3.

Group	Number of Columns	
	CM	Expanded
1	25	25
2	25	25
3	25	25
4	20	23
5	8	25
6	1	25

Table 1: Groups in Problem 3

We note that when the groups are expanded, for the last groups the increase in number of columns is larger than for the first few. This is to be expected for most sparsity structures by the way the methods of partitioning the columns work.

We also considered an expansion of the groups of columns beyond mutual orthogonality to the case where the normal equations for the least-squares subproblems are banded. In particular, we used the following sequential heuristic algorithm to group the columns so that $A_i^T A_i$ is tridiagonal. Initially, the columns are ordered according to some criteria like the incidence degree ordering. Choose the first column, mark it, and let all other columns be unmarked. This will be our first column in the group. Choose the first of the unmarked columns that have a nonzero element in a same row position as the first column and mark all columns that have an element in any same rowposition as the first column. This new

column is our next column in our group. This process is now repeated until a column has no unmarked columns with an element in any same rowposition. At this point, either all columns are marked or there are an unmarked column. Choose the first unmarked column and continue the process until all columns are marked. We have now generated one group of columns so that the the normal matrix $A_i^T A_i$ is block tridiagonal. Columns in different blocks in the same group are orthogonal. Unmark all columns except the columns already in a group. Repeat the process by chosing the first unmarked column. We illustrate this grouping strategy on Problem 3.

Group	Number of columns	blocks
1	45	5
2	41	9
3	17	11
4	1	1

Table 2: Groups in Problem 3

4.3. Storage Requirements

It is of interest to compare the storage requirements of the algorithm of Section 2 applied directly to these problems to the requirements of a very good package for sparse symmetric and positive-definite systems applied to the normal equations. In the following table, columns A and $A^T A$ give the storage required for the real nonzero elements in A and the lower triangular part of $A^T A$, as well as the associated integer pointers when we use the storage scheme of the Harwell testbed. Column L gives the storage requirements for the Yale Sparse Matrix Package (YSMP) [Eisenstat et al (1982)] to store the lower triangular factor of $A^T A$.

Storage Requirement								
Problem	m	n	A		$A^T A$		L	
			Real	Int	Real	Int	Real	Int
1	219	85	438	524	304	390	520	642
2	958	292	1916	2209	1250	1543	2568	2497
3	331	104	662	767	435	540	774	812
4	608	188	1216	1405	796	985	1625	1609
5	313	176	1557	1734	1485	1662	1593	1210

Table 3: Storage

For our scheme, if $A_i^T A_i$, $i=1,2,\cdots,g$ are diagonal matrices, we do not need the vector s_i^k explicitly. Instead, when we compute the components of $A_i^T r^{k+(i-1)/g}$, we also compute the components of s_i^k and accumulate the innerproduct $(A_i^T r^{k+(i-1)/g})^T s_i^k$. Hence the only storage that is needed is the original data A, and b (overwritten by $r^{k+i/g}$), and the solution vector x plus some additional pointer storage for the groups. If $A_i^T A_i$, $i=1,2,\cdots,g$ are tridiagonal, then we need the LDL^T factorization of the tridiagonal matrices and the vector s_i^k. For the inexact Gauss Newton method, we need to store the Jacobian matrix. For the nonlinear substitution method, we need only one extra vector of length m if the columns of the Jacobian matrix or its approximant in each group have no elements in the same rowpositions.

4.4. Numerical Experiments

Now we briefly discuss the termination criteria that we use. From the definition of $r^{k+i/g}$ and the choice of s_i^k, we have

$$\|r^{k+i/g}\|_2^2 = \|r^{k+(i-1)/g}\|_2^2 + 2\omega_i(A_i^T r^{k+(i-1)/g})^T s_i^k + \omega_i^2(A_i^T A_i s_i^k)^T s_i^k$$
$$= \|r^{k+(i-1)/g}\|_2^2 + \omega_i(2-\omega_i)(A_i^T r^{k+(i-1)/g})^T s_i^k.$$

The major work required to calculate the l_2-norm of the residual is an extra innerproduct since $A_i^T r^{k+(i-1)/g}$ is already computed as in Subsection 4.3. Since we want to compare different algorithms, we need to base our stopping criteria on a monotonically decreasing sequence. This suggests the following termination rule:

$$\frac{\|r^{k+i/g}\|_2}{\|r^0\|_2} \leq \varepsilon. \tag{4.1}$$

In the inexact Gauss Newton method, we base the termination rule on the residual $A^T r$ in the normal equations. We note that this costs one matrix-vector product per iteration. We terminate the inner loop cycle when (3.3) holds. For the nonlinear problems, the outer loop is terminated when

$$\frac{\|F(x^{k+i/g})\|}{\|F(x^0)\|} \leq \varepsilon. \tag{4.2}$$

As explained above, one grouping scheme begins by using CM graph coloring to partition the columns of A, and then we use the heuristic strategy to expand the groups. Further, to terminate the iteration to solve the linear problem, we use (4.1). Table 4 compares the CM grouping to the expanded groups that 'overlap'. The entries in the tables are: in the column marked 'it' for iterations, the numbers $kg+i$ in the notation from Section 2 of diagonal least-squares problems solved; we also include in the column marked 'vup' the total number of variables that were updated. Since the block matrix $A_i^T A_i$ is diagonal, the CM grouping is an point SOR using the CM grouping as the ordering. The number of variables updated is therefore the number of point SOR corrections. In Table 4, we choose $\omega_i \equiv 1$.

Number of least squares problems solved													
Problem	g	$\varepsilon=.1$				$\varepsilon=.01$				$\varepsilon=.001$			
		Overlap		CM		Overlap		CM		Overlap		CM	
		it	vup	it	vup	it	vup	it	vup	it	vup	it	vup
1	4	8	188	8	170	19	449	19	408	35	825	35	727
2	6	9	633	9	505	22	1524	23	1166	39	2713	41	2042
3	6	9	223	9	179	28	690	28	511	86	2122	86	1506
4	6	9	411	9	321	24	1092	31	987	52	2362	68	2160
5	10	33	703	33	602	123	2584	123	2186	242	5051	235	4160

Table 4: Overlap vs CM.

Table 4 indicates a fast decrease in the residual in the first few iterations. This can be explained from the observation that the iterative process is somewhat related to coordinate search for the least-squares problem, where the spans of the A_i act like coordinates. Notice

further, there is basically no difference in Table 4, where $\omega=1$, between partitioning the columns and allowing overlaps if we count the number of iterations. We see a bigger difference between overlapping and partitioning for $\omega\neq1$ than for $\omega=1$. Perhaps this can be explained from the observation that if the column a_t of A is in group i and $i+1$, then for $\omega=1$, component t of $\overline{s}_{i+1}^k$ is 0, but for $\omega\neq1$, this component can be nonzero. However, in terms of equivalent point SOR (or point Gauss Seidel) corrections, we see that for the linear problem it does not pay to expand the groups. On the other hand, in the following three sets of results for the nonlinear problems and the nonlinear substitution technique, we see that overlap may be more efficient in terms of fewer iterations and function calls. Of course, this is hardly surprising, but the extra function calls used by nonlinear substitution make it less attractive than the inexact Gauss-Newton method, unless storage is the main concern.

Problem 1

		$\varepsilon=.1$	$\varepsilon=.01$	$\varepsilon=.001$
CM ordering				
	Nonlinear substitution	19/9	45/22	81/40
	Inexact Gauss Newton	11/19/2	21/51/4	26/70/5
Overlap				
	Nonlinear substitution	19/9	45/22	81/40
	Inexact Gauss Newton	11/19/2	21/51/4	26/70/5

Problem 2

		$\varepsilon=.1$	$\varepsilon=.01$	$\varepsilon=.001$
CM ordering				
	Nonlinear substitution	23/11	63/31	105/52
	Inexact Gauss Newton	15/24/2	22/44/3	36/90/5
Overlap				
	Nonlinear substitution	23/11	57/28	103/51
	Inexact Gauss Newton	15/22/2	29/61/4	36/84/4

Problem 3

	$\varepsilon=.1$	$\varepsilon=.01$	$\varepsilon=.001$
CM ordering			
Nonlinear substitution	27/13	57/28	115/57
Inexact Gauss Newton	15/23/2	29/85/4	36/84/5
Overlap			
Nonlinear substitution	23/11	55/27	101/50
Inexact Gauss Newton	15/22/2	29/84/4	36/124/5

f/i/o ,f=number of function call, i=number of iterations, o=number of outer iterations

Table 5: Results for Nonlinear Test Problems

In the following tables, we compare the CM grouping strategy and a grouping strategy so that the corresponding block in the normal equations is block tridiagonal. The strategies are compared to point SOR with the original ordering. The entries for the grouping schemes are the numbers of variables updated to achieve the specified accuracy. For point SOR with the original ordering, they are the numbers of variable updates needed to achieve the same accuracy as the grouping scheme. The arithmetic needed by CM and point SOR with original ordering has the same cost. Naturally, the tridiagonal case costs more per variable update. However, the dominating cost for all the methods is the two matrix-vector products for each sweep through all the columns . The final line in the tables is the relative efficiency in point SOR corrections of the two grouping strategies, i.e. for Problem 1 with $\omega=1.1$ and $\varepsilon=.1$ it is $(191/162)/(171/153)=1.05$.

Point SOR with CM Ordering vs. Point SOR with Original Column Ordering

and

Tridiagonal Blocks vs. Point SOR with Original Ordering

Problem 1

$\varepsilon=.1$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	281	217	196	170	153	170	196	238	302	387	557
Point SOR	304	238	223	210	171	210	237	275	346	434	596
Tridiagonal	255	247	210	170	162	162	210	247	295	417	587
Point SOR	296	268	238	214	191	208	269	288	346	463	625
Rel. eff.	1.07	.99	1.0	1.02	1.05	1.04	1.06	1.01	1.02	.99	1.0

Problem 2

$\varepsilon=.1$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	797	584	559	505	505	559	658	851	1089	1381	1965
Point SOR	802	634	568	552	560	664	807	985	1234	1535	2119
Tridiagonal	814	584	577	522	522	577	705	814	1106	1398	1982
Point SOR	828	664	630	569	613	747	848	975	1277	1557	2164
Rel. eff.	1.01	1.05	1.12	1.0	1.06	1.09	.98	1.03	1.02	1.00	1.01

Problem 3

$\varepsilon=.1$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	283	233	199	179	179	207	233	303	387	491	699
Point SOR	271	233	197	198	202	230	278	329	426	537	745
Tridiagonal	294	253	190	190	190	190	253	294	398	502	710
Point SOR	297	262	198	205	232	222	305	322	436	540	757
Rel. eff.	1.05	1.04	1.05	.98	1.08	1.05	1.01	1.01	1.0	.98	1.00

Problem 4

$\varepsilon=.1$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	548	423	360	321	321	360	468	548	697	924	1261
Point SOR	521	431	341	339	360	387	529	612	762	984	1322
Tridiagonal	529	453	373	341	341	373	453	529	717	905	1281
Point SOR	513	474	365	343	360	461	513	597	799	988	1360
Rel. eff.	1.02	1.03	1.03	.95	.94	1.15	1.00	1.01	1.02	1.03	1.01

Problem 5

$\varepsilon=.1$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	876	778	667	602	602	602	602	654	778	973	1306
Point SOR	802	736	614	583	589	614	633	682	816	1016	1351
Tridiagonal	824	704	621	562	562	562	562	648	738	973	1294
Point SOR	805	715	612	583	588	613	631	700	814	1028	1405
Rel. eff.	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.04	1.05	1.01	1.05

In the next group of tables, we tried point SOR in every case, but the orderings used were the orderings suggested by the column grouping schemes discussed previously. Except for Problem 5, it appears as though the alternate orderings are very good ones. The rows labeled 'Point SOR' correspond to the original ordering.

Point SOR with CM Ordering vs. Point SOR with Original Ordering

and

Point SOR with Tridiagonal Ordering vs. Point SOR with Original Ordering

Problem 1

$\varepsilon=.001$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	1598	1258	961	727	557	493	536	680	897	1173	1683
Point SOR	1870	1530	1273	1020	850	680	610	762	977	1277	1747
Tri	1400	1145	935	765	635	465	550	672	890	1182	1692
Point SOR	1906	1530	1273	1020	903	735	677	777	982	1281	1756

Problem 2

$\varepsilon=.001$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	4162	3286	2603	2042	1606	1534	1898	2410	3066	4162	5815
Point SOR	4381	3505	2886	2311	1850	1875	2268	2751	3377	4389	6009
Tri	3734	3041	2457	1982	1581	1398	1873	2329	3041	4026	5778
Point SOR	4360	3563	2899	2317	2020	1783	2299	2743	3448	4344	6068

Problem 3

$\varepsilon=.001$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	2175	1947	1714	1506	1298	1115	927	832	1090	1447	2071
Point SOR	1762	1559	1450	1247	1141	1034	926	939	1224	1560	2184
Tri	1438	1230	1039	918	773	669	669	831	1085	1438	2062
Point SOR	1763	1560	1352	1349	1142	1038	933	972	1224	1560	2186

Problem 4

$\varepsilon=.001$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	3476	2992	2536	2160	1864	1488	1220	1551	1972	2631	3744
Point SOR	2821	2316	1881	1676	1496	1308	1354	1725	2138	2797	3911
Tri	2521	2145	1769	1469	1205	1017	1205	1501	1957	2629	3725
Point SOR	2823	2388	1960	1685	1488	1309	1448	1709	2159	2851	3891

Problem 5

$\varepsilon=.001$ ω	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
CM	7602	6156	5040	4160	3418	2834	2306	2268	2674	3418	4688
Point SOR	7126	5744	4811	4133	3632	3456	3280	3095	2959	3663	4861
Tri	8306	7384	6574	5842	5138	4434	3758	3112	2702	3464	4962
Point SOR	7125	5744	4812	4133	3684	3456	3280	3095	2959	3663	4855

References

Å. Björk and T. Elfving (1979), Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations, *BIT*, 19, pp.145-163.

T.F.Coleman (1984), *Large Sparse Numerical Optimization*, Lecture Notes in Computer Science 165, Springer-Verlag, Berlin.

T.F.Coleman and J.J.Moré (1982), Software for estimating sparse Jacobian matrices, Cornell Computer Science TR 82-502.

T.F.Coleman and J.J.Moré (1983), Estimation of sparse Jacobian matrices and graph coloring problems, *SIAM J. Numer. Anal.* 20, pp.187-209.

A. R. Curtis, M. J. D. Powell, and J. K. Reid (1974), On the estimation of sparse Jacobian matrices, *J. Inst. Math. Appl.*, 13, pp. 117-119.

R. S. Dembo, S. C. Eisenstat, and T. Steihaug (1982), Inexact Newton methods, *SIAM J. of Numer. Anal.*, 19, pp.400-408.

J. E. Dennis (1977), Nonlinear least squares and equations, *The State of the Art in Numerical Analysis*, D. Jacobs, ed., Academic Press, London, pp.269-312.

J. E. Dennis and R. B. Schnabel (1979), Least change secant updates for quasi-Newton methods, *SIAM Rev.*, 21, pp.443-459.

I.S. Duff and J. K. Reid (1979), Performance Evaluation of Codes for sparse matrices,in *Performance Evaluation of Numerical Software*, L.D.Fosdick (ed.) North Holland Publishing Company, pp.121-135.

S.C. Eisenstat, M.C. Gursky, M.H. Schultz, A.H. Sherman (1982), Yale Sparse Matrix Package I: The symmetric Codes, *International Journal for Numerical Methods in Engineering*, 18, pp.1141-1151.

T. Elfving (1980), Block iterative methods for consistent and inconsistent linear systems, *Numer. Math.*, 35, pp.1-12.

A. S. Householder (1964), *The Theory of Matrices in Numerical Analysis*, Blaisdell Publishing Company, New York.

A. S. Householder and F. L. Bauer (1960), On certain iterative methods for solving linear systems, *Numer. Math.*, 2, pp.55-59.

H. B. Keller (1965), On the solution of singular and semidefinite linear systems by iteration, *SIAM J. of Numer. Math.*, 2, pp.281-290.

J. M. Ortega (1972), *Numerical Analysis: A Second Course*, Academic Press, New York.

J.K.Reid (1973), Least squares solution of sparse systems of non-linear equations by a modified Marquardt algorithm, in *Proc. NATO Conf. at Cambridge, July 1972*, North Holland, Amsterdam, pp.437-445.

T. Steihaug (1980), *Quasi-Newton Methods for Large Scale Nonlinear Problems*, Ph.D dissertation, SOM Technical Report #49, Yale University.

G. W. Stewart (1973), *Introduction to Matrix Computations*, Academic Press, New York.

D. M. Young (1971), *Iterative Solution of Large Linear Systems*, Academic Press, New York.